AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1.　　(Currently Amended)　　A method for managing files in a file system, comprising:

receiving data for a file;

breaking the data in the file into a plurality of segments;

generating an index associated with the file indicating how the file data maps to the segments;

receiving an Input/Output request with respect to a requested address in the file;

using the index associated with the file to determine the segment including data at the requested address in the file;

accessing the determined segment including the data at the requested address;

storing the segments in a primary storage;

copying at least one of the segments in the primary storage onto a secondary storage; and

releasing at least one of the segments copied <u>from the primary storage when copied</u> to the secondary storage, wherein space used by the released segment in the primary storage is available for use <u>and the at least one segment that was copied is not released from the secondary storage</u>.

2. (Original) The method of claim 1, wherein data is stored in the segments by:

writing the received file data to one segment; and

writing further received data for the file to subsequent segments if the last segment to which the received data was written has no more available space.

3. (Original) The method of claim 1, wherein each segment has a fixed byte length, wherein the index provides a segment order indicating an order in which file data is written to the segments, and wherein the index for the file is used to determine the segment including data at the requested address in the file by:

determining an offset into the file including the data at the requested address; and

determining an integer quotient value resulting from the offset into the file divided by the fixed byte length, wherein the segment including the data at the requested address is the segment at the integer quotient value in the segment order.

4. (Original) he method of claim 3, further comprising:

receiving user input indicating the fixed byte length of each segment.

5. (Original) he method of claim 1, further comprising:

providing a segment size that is at least greater than a byte size of a largest section within the file; and

writing each file section to one segment.

6.   (Canceled)

7.   (Previously Presented)   The method of claim 1, wherein as a result of releasing one or more segments, different segments for one file are capable of being stored in the primary storage and the secondary storage.

8.   (Previously Presented)   The method of claim 1, wherein accessing the determined segment including the requested address further comprises:

determining whether the determined segment is available in the primary storage; and

copying the determined segment from the secondary storage to the primary storage if the determined segment is not available in the primary storage.

9.   (Previously Presented)   The method of claim 1, wherein releasing the segment comprises:

storing a partial version of the released segment including less than all data in the segment, wherein the segment data not in the partial version is stored in the secondary storage, wherein the partial version remains on the primary storage after the segment is released.

10. (Original)   The method of claim 9, wherein the partial version of the determined segment is on the primary storage and wherein accessing the determined segment including the requested address further comprises:

accessing the partial version of the determined segment on the primary storage to access the data therein;

reaching the end of the partial version when accessing data therein;

staging from the secondary storage to the primary storage data from the determined segment that is not in the partial version; and

accessing the data from the determined segment staged from the secondary storage to the primary storage.

11. (Original)   The method of claim 9, wherein the partial version is stored only for a first segment of the segments associated with the file.

12. (Previously Presented)   The method of claim 1, further comprising:

accessing data at the end of the segment, wherein the I/O request requires further file data after accessing the end of the segment;

determining from the index a next segment including file data following the file data at the end of the segment data; and

accessing the next segment in the primary storage to access the further required file data.

13. (Previously Presented)   The method of claim 1, further comprising:

maintaining metadata for each segment that is also maintained for files in the file system; and

using the metadata for segments and files to determine when to copy segments and files to the secondary storage and when to release segments and files in the primary storage.

14.   (Original)   The method of claim 13, wherein segments and files in the primary storage are released according to their metadata if used space in the primary storage reaches a threshold level.

15.   (Previously Presented)   The method of claim 1, wherein the file data in all the segments for the file is capable of being larger than a storage capacity of the primary storage.

16.   (Previously Presented)   The method of claim 1, further comprising:

reading data from one target segment on the secondary storage;

determining whether a stage attribute is specified indicating a number of segments to stage ahead; and

initiating read requests to stage the number of subsequent segments following the target segment from the secondary storage to the primary storage.

17.   (Original)   The method of claim 16, further comprising:

receiving user input indicating the number of segments to stage ahead.

18.   (Original)   The method of claim 1, wherein the segment does not have

a file name and is not represented as a file in the file system.

19.   (Original)   The method of claim 1, wherein the index is stored in the file,

wherein no user data is stored in the file and all the user data is distributed in the

segments.

20 - 23.   (Canceled)

24.   (Currently Amended)   A system for managing files, comprising:

a computer readable medium;

a primary storage system;

a secondary storage;

means for receiving data for a file;

means for storing the data for the file in a plurality of segments in the primary

storage system;

means for generating an index in the computer readable medium associated with

the file indicating how the file data maps to the segments;

means for receiving an Input/Output request with respect to an address in the

file;

means for using the index for the file to determine the segment including data at

the requested address in the file;

means for accessing the determined segment including the data at the requested address; ~~and~~

means for copying at least one of the segments in the primary storage onto the secondary storage; and

means for releasing at least one of the segments copied <u>from the primary storage when copied</u> to the secondary storage, wherein space used by the released segment in the primary storage is available for use <u>and the at least one segment that was copied is not released from the secondary storage</u>.

25.   (Previously Presented)   The system of claim 24, wherein the means for storing the data for the file in the segments performs:

writing the received file data to one segment; and

writing further received data for the file to subsequent segments if the last segment to which the received data was written has no more available space.

26.   (Original)   The system of claim 24, wherein each segment has a fixed byte length, wherein the index provides a segment order indicating an order in which file data is written to the segments, and wherein means for using the index for the file to determine the segment including data at the requested address in the file performs:

determining an offset into the file including the data at the requested address; and

determining an integer quotient value resulting from the offset into the file divided by the fixed byte length, wherein the segment including the data at the requested address is the segment at the integer quotient value in the segment order.

27.   (Original)   The system of claim 26, further comprising:

means for receiving user input indicating the fixed byte length of each segment.

28.   (Original)   The system of claim 24, further comprising:

means for providing a segment size that is at least greater than a byte size of a largest section within the file; and

means for writing each file section to one segment.

29.   (Canceled)

30.   (Previously Presented)   The system of claim 24, wherein as a result of releasing one or more segments, different segments for one file are capable of being stored in the primary storage and the secondary storage.

31.   (Previously Presented)   The system of claim 24, wherein the means for accessing the determined segment including the requested address further performs:

determining whether the determined segment is available in the primary storage; and

copying the determined segment from the secondary storage to the primary

storage if the determined segment is not available in the primary storage.

32.   (Previously Presented)   The system of claim 24, wherein the means for

releasing the segment performs:

storing a partial version of the released segment including less than all data in

the segment, wherein the segment data not in the partial version is stored in the

secondary storage, wherein the partial version remains on the primary storage after the

segment is released.

33.   (Original)   The system of claim 32, wherein the partial version of the

determined segment is on the primary storage and wherein the means for accessing the

determined segment including the requested address further performs:

accessing the partial version of the determined segment on the primary storage

to access the data therein;

reaching the end of the partial version when accessing data therein;

staging from the secondary storage to the primary storage data from the

determined segment that is not in the partial version; and

accessing the data from the determined segment staged from the secondary

storage to the primary storage.

34.   (Original)   The system of claim 32, wherein the partial version is stored

only for a first segment of the segments associated with the file.

35.   (Previously Presented)   The system of claim 24, further comprising:

means for accessing data at the end of the segment, wherein the I/O request

requires further file data after accessing the end of the segment;

means for determining from the index a next segment including file data following

the file data at the end of the segment data; and

means for accessing the next segment in the primary storage to access the

further required file data.

36.   (Previously Presented)   The system of claim 24, further comprising:

means for maintaining metadata for each segment that is also maintained for

files in the file system; and

means for using the metadata for segments and files to determine when to copy

segments and files to the secondary storage and when to release segments and files in

the primary storage.

37.   (Original)   The system of claim 24, wherein segments and files in the

primary storage are released according to their metadata if used space in the primary

storage reaches a threshold level.

38.   (Previously Presented)   The system of claim 24, wherein the file data in

all the segments for the file is capable of being larger than a storage capacity of the

primary storage.

39.   (Previously Presented)   The system of claim 24, further comprising:

means for reading data from one target segment on the secondary storage;

means for determining whether a stage attribute is specified indicating a number

of segments to stage ahead; and

means for initiating read requests to stage the number of subsequent segments

following the target segment from the secondary storage to the primary storage.

40.   (Original)   The system of claim 39, further comprising:

means for receiving user input indicating the number of segments to stage

ahead.

41.   (Original)   The system of claim 24, wherein the segment does not have

a file name and is not represented as a file in the file system.

42.   (Original)   The system of claim 24, wherein the index is stored in the

file, wherein no user data is stored in the file and all the user data is distributed in the

segments.

43 - 46. (Canceled)

47.   (Currently Amended)   An article of manufacture containing instructions for managing files in a file system, the instructions being capable of causing a processor to:

receive data for a file;

break the data in the file into a plurality of segments;

generate an index associated with the file indicating how the file data maps to the segments;

receive an Input/Output request with respect to a requested address in the file;

use the index associated with the file to determine the segment including data at the requested address in the file;

access the determined segment including the data at the requested address;

store the segments in a primary storage;

copy at least one of the segments in the primary storage onto a secondary storage; and

release at least one of the segments copied from the primary storage when copied to the secondary storage, wherein space used by the released segment in the primary storage is available for use and the at least one segment that was copied is not released from the secondary storage.

48.   (Previously Presented)   The article of manufacture of claim 47, wherein the processor is further operable to:

write the received file data to one segment; and

write further received data for the file to subsequent segments if the last segment

to which the received data was written has no more available space.

49.   (Previously Presented)   The article of manufacture of claim 47, wherein

each segment has a fixed byte length, wherein the index provides a segment order

indicating an order in which file data is written to the segments, and wherein the index

for the file is used to determine the segment including data at the requested address in

the file by the processor being further operable to:

determine an offset into the file including the data at the requested address; and

determine an integer quotient value resulting from the offset into the file divided

by the fixed byte length, wherein the segment including the data at the requested

address is the segment at the integer quotient value in the segment order.

50.   (Previously Presented)   The article of manufacture of claim 49, wherein

the processor is further operable to:

receive user input indicating the fixed byte length of each segment.

51.   (Previously Presented)   The article of manufacture of claim 47, wherein

the processor is further operable to:

provide a segment size that is at least greater than a byte size of a largest

section within the file; and

write each file section to one segment.

52.   (Canceled)

53.   (Previously Presented)   The article of manufacture of claim 47, wherein as a result of releasing one or more segments, different segments for one file are capable of being stored in the primary storage and the secondary storage.

54.   (Previously Presented)   The article of manufacture of claim 47, wherein the processor is further operable to:

determine whether the determined segment is available in the primary storage; and

copy the determined segment from the secondary storage to the primary storage if the determined segment is not available in the primary storage.

55.   (Currently Amended)   The article of manufacture of claim 47, wherein the processor is further operable to:

store a partial version of the released segment including less than all data in the segment, wherein the segment data not in the partial version is stored in the secondary storage, wherein the partial version remains on the primary storage after the segment is released.

56.   (Currently Amended)   The article of manufacture of claim 55, wherein the partial version of the determined segment is on the primary storage and wherein the processor is further operable to:

access the partial version of the determined segment on the primary storage to access the data therein;

reach the end of the partial version when accessing data therein;

stage from the secondary storage to the primary storage data from the determined segment that is not in the partial version; and

<u>access</u> the data from the determined segment staged from the secondary storage to the primary storage.

57.   (Original)   The article of manufacture of claim 55, wherein the partial version is stored only for a first segment of the segments associated with the file.

58.   (Previously Presented)   The article of manufacture of claim 47, wherein the processor is further operable to:

access data at the end of the segment, wherein the I/O request requires further file data after accessing the end of the segment;

determine from the index a next segment including file data following the file data at the end of the segment data; and

access the next segment in the primary storage to access the further required file data.

59.   (Previously Presented)   The article of manufacture of claim 47, wherein the processor is further operable to:

maintain metadata for each segment that is also maintained for files in the file

system; and

use the metadata for segments and files to determine when to copy segments

and files to the secondary storage and when to release segments and files in the

primary storage.

60. (Original) The article of manufacture of claim 59, wherein segments

and files in the primary storage are released according to their metadata if used space

in the primary storage reaches a threshold level.

61. (Previously Presented) The article of manufacture of claim 47, wherein

the file data in all the segments for the file is capable of being larger than a storage

capacity of the primary storage.

62. (Previously Presented) The article of manufacture of claim 47, wherein

the processor is further operable to:

read data from one target segment on the secondary storage;

determine whether a stage attribute is specified indicating a number of segments

to stage ahead; and

initiate read requests to stage the number of subsequent segments following the

target segment from the secondary storage to the primary storage.

63.　(Previously Presented)　The article of manufacture of claim 62, wherein the processor is further operable to:

receive user input indicating the number of segments to stage ahead.

64.　(Original)　The article of manufacture of claim 47, wherein the segment does not have a file name and is not represented as a file in the file system.

65.　(Original)　The article of manufacture of claim 47, wherein the index is stored in the file, wherein no user data is stored in the file and all the user data is distributed in the segments.

66 - 69.　(Canceled)